On the solution of the heat equation in very thin tapes

E. Prulière [a,*], F. Chinesta [b], A. Ammar [c], A. Leygue [b], A. Poitou [b]

[a] Institut de Mécanique et d'Ingénierie de Bordeaux, 16 avenue Pey Berland, 33607 PESSAC Cedex, France
[b] GEM UMR CNRS, Ecole Centrale de Nantes, EADS Corporate Foundation International Chair, 1 rue de la Noe, BP 92101, F-44321 Nantes cedex 3, France
[c] Arts et Métiers ParisTech, 2 Boulevard du Ronceray, BP 93525, F-49035 Angers cedex 01, France

ARTICLE INFO

Article history:
Received 14 October 2011
Received in revised form
7 September 2012
Accepted 9 October 2012
Available online 7 December 2012

Keywords:
Heat equation
Model reduction
Proper generalized decomposition
Composites manufacturing processes

ABSTRACT

This paper addresses two issues usually encountered when simulating thermal processes in forming processes involving tape-type geometries, as is the case of tape or tow placement, surface treatments, ⋯ The first issue concerns the necessity of solving the transient model a huge number of times because the thermal loads are moving very fast on the surface of the part and the thermal model is usually non-linear. The second issue concerns the degenerate geometry that we consider in which the thickness is usually much lower than the in-plane characteristic length. The solution of such 3D models involving fine meshes in all the directions becomes rapidly intractable despite the huge recent progresses in computer sciences. In this paper we propose to consider a fully space-time separated representation of the unknown field. This choice allows circumventing both issues allowing the solution of extremely fine models very fast, sometimes in real time.

1. Introduction

Industrial processes generally need efficient numerical simulations in order to optimize the process parameters. In the case of composite materials, even if the thermo-mechanical models are nowadays well established, efficient simulations need for further developments.

In this work we are considering some issues, analyzed from a methodological point of view, without considering its industrial counterpart that requires the coupling of different numerical procedures and richer physics.

Thermal models involved in the numerical modeling of composite tape placement processes introduce, despite its geometrical simplicity, a certain number of numerical difficulties related to: (i) the very fine mesh required due to the small domain thickness with respect to the other characteristic dimensions as well as to the presence of a thermal source moving on the domain surface; and (ii) the long simulation times induced by the low thermal conductivity of polymers and the movement of the heat source;

The solution by using standard discretization techniques can be extremely expensive from the computing time point of view. For example, if one wants to simulate a thermal problem in a ply whose thickness is 1000 times lower than its length (which is a quite common ratio), the use of only 100 nodes in the thickness will lead to use 10^5 nodes in the length to ensure the geometrical quality of the mesh on which standard discretization techniques, like the finite element method, proceed. The total amount of nodes is then 10 millions even when considering a 2D thermal model. In this situation solving a 3D model seems a challenge. Indeed, when the model involves 10^{12} (that implies a reasonable number of nodes, of the order of 10^4 in each coordinate direction of a 3D model) numerical complexity reaches the current computer capabilities. In addition, in transient non-linear models the problem must be solved at least once at each time step, time step that can be extremely small due to stability constraints.

In order to reduce the computing time needed for solving large numerical models, different ways have been explored. One consists in using super high performance computing facilities. Others strategies consider subdomains, multigrid techniques or the use of efficient preconditioners. Another efficient way to enhance the simulation capabilities is to reduce the size of the approximation basis employed for approximating the unknown field. In the finite elements method, at least one approximation function is associated to each node. Thus, the number of degrees of freedom scales with the number of nodes. Reduced modeling lies in using a reduced number of "appropriate" approximation functions defined in general in the whole domain and able to approximate up to a certain level of accuracy the problem solution at each time. Thus, the number of approximation functions (and by the way the

* Corresponding author.
E-mail addresses: etienne.pruliere@lamef.bordeaux.ensam.fr (E. Prulière), Francisco.Chinesta@ec-nantes.fr (F. Chinesta), Amine.Ammar@ensam.eu (A. Ammar), Adrien.Leygue@ec-nantes.fr (A. Leygue), Arnaud.Poitou@ec-nantes.fr (A. Poitou).

1290-0729/$ – see front matter
http://dx.doi.org/10.1016/j.ijthermalsci.2012.10.017

number of degrees of freedom) becomes independent of the mesh size. The arising issue is how to calculate these "appropriate" functions defining the reduced approximation basis?

There are several possibilities. A first possibility lies in the use of the Proper Orthogonal Decomposition – POD – that was employed in a former work [9] for addressing similar issues to the ones concerned by the present work. In what follows we are describing how the POD extracts relevant information for building-up a reduced approximation basis.

1.1. Extracting relevant information by applying the proper orthogonal decomposition

We assume that the field of interest $u(\mathbf{x},t)$ is known at the nodes $\mathbf{x}_i$ of a spatial mesh for discrete times $t_m = m\cdot\Delta t$, with $i\in[1,\cdots,M]$ and $m\in[0,\cdots,P]$. We use the notation $u(\mathbf{x}_i,t_m)\equiv u^m(\mathbf{x}_i)\equiv u_i^m$ and define $\mathbf{u}^m$ as the vector of nodal values u_i^m at time t_m. The main objective of the POD is to obtain the most typical or characteristic structure $X(\mathbf{x})$ among these $u^m(\mathbf{x})$, $\forall m$. For this purpose, we solve the following eigenvalue problem [32]:

$$\mathbf{C}X = \alpha\mathbf{X}. \tag{1}$$

Here, the components of vector $\mathbf{X}$ are $X(\mathbf{x}_i)$, and $\mathbf{C}$ is the two-point correlation matrix

$$\mathbf{C}_{ij} = \sum_{m=1}^{P} u^m(\mathbf{x}_i)\cdot u^m(\mathbf{x}_j), \tag{2}$$

whose matrix form reads:

$$\mathbf{C} = \sum_{m=1}^{P} \mathbf{u}^m\cdot(\mathbf{u}^m)^T, \tag{3}$$

which is symmetric and positive definite. With the matrix $\mathbf{Q}$ defined as

$$\mathbf{Q} = \left(\mathbf{u}^1,\cdots,\mathbf{u}^P\right) \tag{4}$$

We have

$$\mathbf{C} = \mathbf{Q}\cdot\mathbf{Q}^T. \tag{5}$$

1.2. Building the POD reduced-order model

In order to obtain a reduced model, we first solve the eigenvalue problem Eq. (1) and select the N eigenvectors $\mathbf{X}_i$, $i = 1,\cdots,N$, associated with the N eigenvalues belonging to the interval defined by the highest eigenvalue α_1 and α_1 divided by a large enough number (e.g. 10^8). In practice, N is found to be much lower than M. These N eigenfunctions $\mathbf{X}_i$ are then used to approximate the solution $u^m(\mathbf{x})$, $\forall m$. To this end, let us define the matrix $\mathbf{B} = (\mathbf{X}_1\cdots\mathbf{X}_N)$.

Now, let us assume for illustrative purposes that an explicit time-stepping scheme is used to compute the discrete solution $\mathbf{u}^{m+1}$ at time t^{m+1}. One must thus solve a linear algebraic system of the form

$$\mathbf{G}^m\mathbf{u}^{m+1} = \mathbf{H}^m. \tag{6}$$

A reduced-order model is then obtained by approximating $\mathbf{u}^{m+1}$ in the subspace defined by the N eigenvectors $\mathbf{X}_i$, i.e.

$$\mathbf{u}^{m+1}\approx\sum_{i=1}^{N}\mathbf{X}_i\cdot T_i^{m+1} = \mathbf{B}\cdot\mathbf{T}^{m+1}. \tag{7}$$

Eq. (6) then reads

$$\mathbf{G}^m\cdot\mathbf{B}\cdot\mathbf{T}^{m+1} = \mathbf{H}^m, \tag{8}$$

or equivalently

$$\mathbf{B}^T\cdot\mathbf{G}^m\cdot\mathbf{B}\cdot\mathbf{T}^{m+1} = \mathbf{B}^T\cdot\mathbf{H}^m. \tag{9}$$

The coefficients $\mathbf{T}^{m+1}$ defining the solution of the reduced-order model at the time step $m+1$ are thus obtained by solving an algebraic system of size N instead of M. When $N \ll M$, as is the case in numerous applications, the solution of Eq. (9) is thus preferred because of its much reduced size.

Remark 1 The reduced-order model Eq. (9) is built a posteriori by means of the already-computed discrete field evolution. Thus, one could wonder about the interest of the whole exercise. In fact, two beneficial approaches are widely considered (see e.g. [6,8,18,24–26,31,32]). The first approach consists in solving the large original model over a short time interval, thus allowing for the extraction of the characteristic structure that defines the reduced model. The latter is then solved over larger time intervals, with the associated computing time savings. The other approach consists in solving the original model over the entire time interval, and then using the corresponding reduced model to solve very efficiently similar problems with, for example, slight variations in material parameters or boundary conditions. We considered some years ago an adaptive technique for constructing the reduced basis without an "a priori" knowledge [2,31,32], following the original proposal in [30].

Remark 2 The construction of the reduced bases is not unique. There are many alternatives. Some ones introduce some improvements on the POD methodology just described, as is the case of the Goal Oriented Model Constrained Optimization approach (see [7] and the references therein) or the modal identification method (see [13] and the references therein). The Branch Eigenmodes Reduction Method combined with the amalgam method is another appealing constructor of reduced bases [34].

Remark 3 The application of the POD allows to express the unknown function $u(\mathbf{x},t)$ in the reduced space-time separated form

$$u(\mathbf{x},t)\approx\sum_{i=1}^{i=N}T_i(t)\cdot X_i(\mathbf{x}) \tag{10}$$

where $X_i(\mathbf{x})$ are space dependent function (the eigenfunctions resulting from the application of the POD) and $T_i(t)$ are its coefficients that only depend on time.

1.3. From POD to PGD

Despite the fact of having proposed techniques able to define the reduced basis without an "a priori" knowledge, the robustness of such strategies is not ensured and in some cases these strategies do not converge. In that case one could consider as starting point a separated representation of the problem solution $u(\mathbf{x},t)$

$$u(\mathbf{x},t)\approx\sum_{i=1}^{i=N}T_i(t)\cdot X_i(\mathbf{x}) \tag{11}$$

and then inject it in the weak form of the problem. This procedure allows computing the functions involved in the separated approximation without any "a priori" knowledge. This strategy was proposed by Pierre Ladeveze in the 80's, and he called it radial approximation [19,20,23].

Inspired by this procedure one could try to generalize this representation to the multidimensional fields as was proposed

in [1,3]. This generalized formulation was called Proper Generalized Decomposition −PGD−. See [10] for a recent review. In the PGD framework, when the domain is hexahedral an appealing separated representation of $u(\mathbf{x},t)$ consists of a full separation, i.e.

$$u(\mathbf{x},t) \approx \sum_{i=1}^{i=N} T_i(t)\cdot X_i(x)\cdot Y_i(y)\cdot Z_i(z) \tag{12}$$

In the case of non hexahedral domains, a fully separated representation is always possible as proved in [16] but it involves some technical points. Otherwise, a restricted separated representation in space and time as in Eq. (11) is always possible and allows to use unstructured meshes on the whole domain.

The solution computed by applying the PGD method is only valid for the model parameters and boundary conditions considered to derive it. So, more than a standard model reduction technique, the space-time separated representation considered above can be viewed as a fast and cheap procedure to compute a sort of "compressed" solution. Obviously as soon as we include model parameters or the ones describing initial or boundary conditions as extra-coordinates (as illustrated in the last section of this paper), the solution of the multi-dimensional resulting problem allows having access to the solution for any choice of those parameters, and in that case PGD can be viewed as a model reduction strategy (if the separated representation is truncated after having calculated some terms), like the reduced models obtained by using POD (many references were given previously) or reduced bases [22,28,33], but it does not need any "a priori" calculation, the reduced model is calculated on the fly. In what follows the PGD will be considered as a direct solver and not as a reduced model, because we are computing all the modes needed to reach a prescribed level of accuracy. However, the space-time solution expressed in a separated form can be viewed as a sort of compressed representation obtained directly during the solution process.

PGD assumes a separated representation of the problem solution. So we should address the issue related to the separability of a problem solution. For the sake of simplicity we are considering a problem involving a scalar 2D solution $u(x,y)$. If we consider the polynomial approximation of function $u(x,y)$

$$u(x,y) = a_{0,0} + a_{1,0}\cdot x + a_{0,1}\cdot y + a_{1,1}\cdot x\cdot y + a_{2,0}\cdot x^2 + \cdots \tag{13}$$

we can notice that it writes in a separated form, that is, sums of products of functions of x multiplied by functions of y.

When the solution can be expressed in a subspace of the full tensor product space generated by the polynomial bases in both coordinates x and y, i.e. in a subspace of (13), the PGD constructor avoids taking into account terms in (13) that not involved in the solution. Obviously, if the solution is non-separable, i.e. it involves all the terms resulting from the one dimensional bases tensor product, the PGD does not present any advantage, because the enrichment procedure continues until defining the whole tensor product basis that corresponds to a solution with the same complexity that the ones computed by using any mesh-based discretization technique. This situation can be found when solving hyperbolic equations or advection dominated advection-diffusion equations. The simplest way for checking the problem solution separability consists of solving the problem by using a standard discretization technique in a mesh of moderate complexity, and then applying the singular value decomposition −SVD− of its multidimensional counterpart, the high order SVD known as HOSVD [16].

In the present paper we are applying a fully separated representation of the temperature field defined in an hexahedral space-time domain on which a thermal source is moving. In the last section we will come back to the parametric modeling issue.

2. Proper generalized decomposition of a thermal model defined in rectangular domain

For the sake of simplicity in the description of the technique we consider the application of the PGD for solving the transient heat equation in a 2D rectangular spatial domain (3D results will be presented later) because its generalization for addressing multidimensional problems is straightforward.

The transient thermal model is defined in $\Omega\times\mathcal{I}$, $\Omega=\Omega_x\times\Omega_y$ ($\Omega_x=(0,L)$ and $\Omega_y=(0,H)$) and $\mathcal{I}=(0,t_{\max}]$. The evolution of the temperature field $u(x,y,t)$ is governed by the heat equation

$$\rho\cdot C_p\cdot\frac{\partial u}{\partial t} - \nabla\cdot(\mathbf{K}\cdot\nabla u) = 0 \tag{14}$$

where $\mathbf{K}$ represents the conductivity tensor, assumed, without loss of generality, constant. If we proceed in the coordinate system associated with the principal directions of $\mathbf{K}$, the conductivity tensor becomes diagonal, being its components k_x and k_y the principal thermal conductivities. In that system of coordinates the previous equation reduced to:

$$\rho\cdot C_p\cdot\frac{\partial u}{\partial t} - k_x\frac{\partial^2 u}{\partial x^2} - k_y\frac{\partial^2 u}{\partial y^2} = 0 \tag{15}$$

We assume, without loss of generality, a constant initial temperature

$$u(x,y,t=0) = u^0 \tag{16}$$

and we prescribe the heat flux on the whole boundary $\Gamma\equiv\partial\Omega$, $\Gamma=\Gamma_1\cup\Gamma_2\cup\Gamma_3\cup\Gamma_4$, $\Gamma_1=(x=0,y\in\Omega_y)$, $\Gamma_2=(x\in\Omega_x,y=0)$, $\Gamma_3=(x=L,y\in\Omega_y)$ and $\Gamma_4=(x\in\Omega_x,y=H)$:

$$\begin{cases} \left.\dfrac{\mathrm{d}u}{\mathrm{d}x}\right|_{\mathbf{x}\in\Gamma_1} = 0 \\ \left.\dfrac{\mathrm{d}u}{\mathrm{d}y}\right|_{\mathbf{x}\in\Gamma_2} = 0 \\ \left.\dfrac{\mathrm{d}u}{\mathrm{d}x}\right|_{\mathbf{x}\in\Gamma_3} = 0 \\ -k_y\cdot\left.\dfrac{\mathrm{d}u}{\mathrm{d}y}\right|_{\mathbf{x}\in\Gamma_4} = \begin{cases} -q(x,t); & x\in\Gamma_4^s \\ h\cdot(u(y=H,t)-u_{\mathrm{amb}}); & x\in\Gamma_4-\Gamma_4^s \end{cases} \end{cases} \tag{17}$$

where $q(x,t)$ represents the heating source that is moving on the upper surface, Γ_4^s being the region on which it applies, h is a convection coefficient describing the heat transfer between the domain Ω and its environment assumed at temperature u_{amb}.

The weak form related to Eq. (15) reads:

$$\int_{\Omega\times\mathcal{I}} u^*\cdot\left(\rho\cdot C_p\cdot\frac{\partial u}{\partial t} - k_x\frac{\partial^2 u}{\partial x^2} - k_y\frac{\partial^2 u}{\partial y^2}\right)\mathrm{d}\Omega\cdot\mathrm{d}t = 0 \tag{18}$$

$\forall u^*$ in an appropriate functional space.

In order to transfer the boundary condition into the integral formulation (18) we perform a spatial integration by parts, which results in

$$\int_{\Omega\times\mathcal{I}} u^*\cdot\rho\cdot C_p\cdot\frac{\partial u}{\partial t}\mathrm{d}\Omega\cdot\mathrm{d}t+\int_{\Omega\times\mathcal{I}} k_x\frac{\partial u^*}{\partial x}\cdot\frac{\partial u}{\partial x}\mathrm{d}\Omega\cdot\mathrm{d}t + \int_{\Omega\times\mathcal{I}} k_y\frac{\partial u^*}{\partial y}\cdot\frac{\partial u}{\partial y}\mathrm{d}\Omega\cdot\mathrm{d}t+\int_{\Gamma_4} u^*\cdot\tilde{q}(x,t)\mathrm{d}x\cdot\mathrm{d}t = 0 \tag{19}$$

where $\tilde{q}(x,t)$ represents $-k_y\cdot\frac{\mathrm{d}u}{\mathrm{d}y}|_{\mathbf{x}\in\Gamma_4}$, that is,

$$\tilde{q}(x,t) = \begin{cases} -q(x,t); & x\in\Gamma_4^s \\ h\cdot(u(y=H,t)-u_{\mathrm{amb}}); & x\in\Gamma_4-\Gamma_4^s \end{cases} \tag{20}$$

Now, we assume a separated representation of the temperature field

$$u(x,y,t)\approx\sum_{i=1}^{i=N} T_i(t)\cdot X_i(x)\cdot Y_i(y) \tag{21}$$

In order to construct such representation we proceed iteratively, by computing a term of the finite sum at each iteration. If we assume that at iteration n, functions $X_i(x)$, $Y_i(y)$ and $T_i(t)$, $i=1,\cdots,n$, were already computed, the solution at iteration n, $u^n(x,y,t)$ writes:

$$u^n(x,y,t) = \sum_{i=1}^{i=n} T_i(t)\cdot X_i(x)\cdot Y_i(y) \tag{22}$$

Remark 4 In order to ensure the verification of the initial condition (the Neunman's boundary ones are implicit in the weak formulation) we could consider that the first term of the finite sum decomposition is given by $T_1(t)=u_0$ and $X_1(x)=Y_1(y)=1$. In more complex situations the interested reader can refer to [16].

At iteration $n+1$ we look for the new functions $X_{n+1}(x)$, $Y_{n+1}(y)$ and $T_{n+1}(t)$ that for the sake of clarity will be denoted by $R(x)$, $S(y)$ and $W(t)$. Thus, we can write:

$$u^{n+1}(x,y,t) = u^n(x,y,t)+R(x)\cdot S(y)\cdot W(t) \tag{23}$$

The associated weighting function u^* reads:

$$u^*(x,y,t) = R^*(x)\cdot S(y)\cdot W(t)+R(x)\cdot S^*(y)\cdot W(t) + R(x)\cdot S(y)\cdot W^*(t) \tag{24}$$

Introducing Eqs. (23) and (24) into Eq. (19) yields a non-linear integral problem because each unknown function ($R(x)$, $S(y)$ and $W(t)$) never appear isolated but is always multiplying several unknown functions. The test functions R^* and S^* are approximated by using the same approximation bases that the ones considered for approximating the trial functions R and S, as in a classical Galerkin framework. However, one must proceed carefully when addressing the time problem because its advective nature. In that case we considered in our former works either an upwind or discontinuous Galerkin time discretization operating on the weak form governing the solution $W(t)$ or a backward finite differences discretization on the strong form that ca be obtained from the weak form taking into account the arbitrariness of function W^*.

Because the nonlinearity, a linearization strategy is compulsory. In our earlier papers [1] and [3], we used Newton's method. Simpler linearization strategies can also be applied. The simplest one is an alternating directions fixed-point algorithm, which was found remarkably robust in the present context. Each iteration consists of three steps that are repeated until reaching convergence, that is, until reaching the fixed point. The first step assumes $S(y)$ and $W(t)$ known from the previous iteration and computes an update for $R(x)$ (in this case the test function reduces to $R^*(x)\cdot S(y)\cdot W(t)$). From the just-updated $R(x)$ and the previously-used $W(t)$, we can update $S(y)$ (with $u^*=R(x)\cdot S^*(y)\cdot W(t)$). Finally, from the just-computed $R(x)$ and $S(y)$, we update $W(t)$ (with $u^*=R(x)\cdot S(y)\cdot W^*(t)$). This iterative procedure continues until convergence. The converged functions define the new term in the expansion (21) of $u(x,y,t)$: $X_{n+1}(x)=R(x)$, $Y_{n+1}(y)=S(y)$ and $T_{n+1}(t)=W(t)$.

In what follows we come back to the problems to be solved at each one of these three steps.

(1) Computing $R(x)$ being $S(y)$ and $W(t)$ given.

In the present case the test function reads:

$$u^* = R^*(x)\cdot S(y)\cdot W(t) \tag{25}$$

that introduced into the integral form (19) results in:

$$\begin{aligned}
&\int_{\Omega_x\times\Omega_y\times\mathcal{I}} R^*\cdot S\cdot W\cdot\rho\cdot C_p\cdot R\cdot S\cdot\frac{\mathrm{d}W}{\mathrm{d}t}\mathrm{d}x\cdot\mathrm{d}y\cdot\mathrm{d}t\\
&+\int_{\Omega_x\times\Omega_y\times\mathcal{I}} k_x\cdot\frac{\mathrm{d}R^*}{\mathrm{d}x}\cdot S\cdot W\cdot\frac{\mathrm{d}R}{\mathrm{d}x}\cdot S\cdot W\mathrm{d}x\cdot\mathrm{d}y\cdot\mathrm{d}t\\
&+\int_{\Omega_x\times\Omega_y\times\mathcal{I}} k_y\cdot R^*\cdot\frac{\mathrm{d}S}{\mathrm{d}y}\cdot W\cdot R\cdot\frac{\mathrm{d}S}{\mathrm{d}y}\cdot W\mathrm{d}x\cdot\mathrm{d}y\cdot\mathrm{d}t\\
&+\int_{\Omega_x\times\mathcal{I}} R^*\cdot S(y=H)\cdot W\cdot\tilde{q}(x,t)\mathrm{d}x\cdot\mathrm{d}t\\
&=-\int_{\Omega_x\times\Omega_y\times\mathcal{I}} R^*\cdot S\cdot W\cdot\rho\cdot C_p\cdot\sum_{i=1}^{i=n}\left(X_i\cdot Y_i\cdot\frac{\mathrm{d}T_i}{\mathrm{d}t}\right)\mathrm{d}x\cdot\mathrm{d}y\cdot\mathrm{d}t\\
&-\int_{\Omega_x\times\Omega_y\times\mathcal{I}} k_x\cdot\frac{\mathrm{d}R^*}{\mathrm{d}x}\cdot S\cdot W\cdot\sum_{i=1}^{i=n}\left(\frac{\mathrm{d}X_i}{\mathrm{d}x}\cdot Y_i\cdot T_i\right)\mathrm{d}x\cdot\mathrm{d}y\cdot\mathrm{d}t\\
&-\int_{\Omega_x\times\Omega_y\times\mathcal{I}} k_y\cdot R^*\cdot\frac{\mathrm{d}S}{\mathrm{d}y}\cdot W\cdot\sum_{i=1}^{i=n}\left(X_i\cdot\frac{\mathrm{d}Y_i}{\mathrm{d}y}\cdot T_i\right)\mathrm{d}x\cdot\mathrm{d}y\cdot\mathrm{d}t
\end{aligned} \tag{26}$$

where the dependences of R, S and W on their respective coordinates were omitted for the sake of clarity.

As all the functions involving the y and t coordinates are known, we can integrate Eq. (26) in $\Omega_y\times\mathcal{I}$ leading to:

$$\int_{\Omega_x}\left(R^*\cdot\alpha^x\cdot R+\frac{\mathrm{d}R^*}{\mathrm{d}x}\cdot\beta^x\cdot\frac{\mathrm{d}R}{\mathrm{d}x}\right)\mathrm{d}x = \int_{\Omega_x}\left(R^*\cdot\gamma^x(x)+\frac{\mathrm{d}R^*}{\mathrm{d}x}\cdot\delta^x(x)\right)\mathrm{d}x \tag{27}$$

where α^x and β^x are two constants and $\gamma^x(x)$ and $\delta^x(x)$ are functions of x. Eq. (27) can be solved by using any standard technique, as for example a 1D finite element discretization.

Note that an efficient implementation requires a separated representation of the thermal source

$$q(x,t)\approx\sum_{i=1}^{i=Q} F_i(x)\cdot G_i(t) \tag{28}$$

decomposition that can be performed as proved later by using the SVD (singular value decomposition).

Note also that integrating by parts the terms involving the derivatives of the test function R^* and then using its arbitrariness one could obtain an equivalent strong form that could be then discretized by using any appropriate technique.

(2) Computing $S(y)$ being $R(x)$ and $W(t)$ given.

In this case and proceeding in a similar way that previously but integrating in $\Omega_x \times \mathcal{I}$ it results in

$$\int_{\Omega_y} \left(S^* \cdot \alpha^y \cdot S + \frac{dS^*}{dy} \cdot \beta^y \cdot \frac{dS}{dy} \right) dy = S^*(y = H) \cdot \xi^y + \int_{\Omega_y} \left(S^* \cdot \gamma^y(y) + \frac{dS^*}{dy} \cdot \delta^y(y) \right) dy \tag{29}$$

In the present case the integral on $\Gamma_4 \times \mathcal{I}$ results in constant value ξ^y.

(3) Computing $W(t)$ being $R(x)$ and $S(y)$ given.

Now, the weak form is integrated in $\Omega_x \times \Omega_y$ to derive the equation given $W(t)$. In the present case it is easy to verify that the resulting equation reads:

$$\int_{\mathcal{I}} W^* \cdot \left(\alpha^t \cdot \frac{dW}{dt} + \beta^t \cdot W \right) dt = \int_{\mathcal{I}} W^* \cdot \delta^t(t) dt \tag{30}$$

One could solve this weak form by using a stabilized discretization technique (e.g. discontinuous Galerkin) or coming back to its strong form

$$\alpha^t \cdot \frac{dW}{dt} + \beta^t \cdot W = \delta^t(t) \tag{31}$$

that can be solved by using any standard finite difference discretization (e.g. backward Euler, among many others).

In the previous steps functions $R(x)$ and $X_i(x)$, $S(y)$ and $Y_i(y)$ and $W(t)$ and $T_i(t)$ are approximated by using an appropriate mesh. After computing the solution, its associated error can be evaluated by using an appropriate error estimator, as the one proposed in [5], and then different choices exist: (i) computing more terms in the finite sums decomposition, or (ii) refine the meshes used for approximating the different trial and test functions involved in the separated approximation. One could think that when considering the last strategy the whole solution should be recomputed, but in fact the coarse solution $u^N(x,y,z,t)$ already calculated

$$u^N(t,x,y) = \sum_{i=1}^{i=N} T_i(t) \cdot X_i(x) \cdot Y_i(y) \tag{32}$$

can be enriched by introducing $\tilde{N}$ additional terms in the sum

$$u(t,x,y) \approx \sum_{i=1}^{i=N} T_i(t) \cdot X_i(x) \cdot Y_i(y) + \sum_{i=1}^{i=\tilde{N}} \tilde{T}_i(t) \cdot \tilde{X}_i(x) \cdot \tilde{Y}_i(y) \tag{33}$$

whose associated functions $\tilde{X}_i$, $\tilde{Y}_i$ and $\tilde{T}_i$ are approximated by employing finer meshes. Despite its conceptual simplicity in what follows we are not considering refinement strategies.

The solution procedure can be summarized as follows:

- $n = 0$
- While $Error_1 > \varepsilon_1$
  - Initialize $S^{(0)}(y)$ and $W^{(0)}(t)$
  - $k = 0$
  - While $Error_2 > \varepsilon_2$
    - $k \leftarrow k + 1$
    - Compute $R^{(k)}$ from Eq. (27)
    - Compute $S^{(k)}$ from Eq. (29)
    - Compute $W^{(k)}$ from Eq. (31)
    - $Error_2^2 = \|R^{(k)} - R^{(k-1)}\|^2 + \|S^{(k)} - S^{(k-1)}\|^2 + \|W^{(k)} - W^{(k-1)}\|^2$
  - $Error_1^2 = \|R^{(k)} \cdot S^{(k)} \cdot W^{(k)}\|^2$ (other stooping criteria exist)
  - $X_{n+1} = R$, $Y_{n+1} = S$ and $T_{n+1} = W$
  - $n \leftarrow n + 1$

We have seen that at each enrichment step the construction of the new functional product in Eq. (21) requires iterations. If m_i denotes the number of iterations needed at enrichment step i for computing $X_i(x)$, $Y_i(y)$ and $T_i(t)$, the total number of iterations involved in the construction of the PGD approximation is $m = \sum_{i=1}^{i=N} m_i$. In the above example, the entire procedure thus involves the solution of $2 \cdot m$ ($3 \cdot m$ in 3D thermal problems) one-dimensional boundary values problems for the functions $X_i(x)$ and $Y_i(y)$ and m one-dimensional initial values problems for the functions $T_i(t)$. In general, m_i rarely exceeds ten. The number N of functional products needed to approximate the solution with enough accuracy depends on the solution regularity. All numerical experiments carried to date reveal that N ranges between a few tens and one hundred. Thus, we can conclude that the complexity of the PGD procedure to compute the approximation (21) is of few hundreds of 1D problems. In a classical approach, one must solve a 2D problem at each time step. In usual applications, this often implies the computation of several millions of 2D solutions. Clearly, the CPU time savings by applying the PGD can be of several orders of magnitude.

Remark 5 The just proposed strategy also applies for solving non-linear models. In that case many standard linearization strategies can be considered. Thus, one could expect that when looking for the solution at iteration $n+1$, u^{n+1}, all the non-linear terms could be considered at the previous iteration, by using u^n for evaluating all the non-linear contributions. This technique runs, as well as many other variants [4]. A non-conventional and specially appealing technique for addressing complex non linearities lies in the use of the LATIN method [19,20,23].

Remark 6 Because when using the PGD method the computational complexity scales linearly with the model dimensionality instead of the exponential growing characteristic of mesh based discretization techniques, one could introduce new extra-coordinates in the model, other than the usual space and time, without a significant impact on the CPU time. Thus, thermal parameters, initial and/or boundary conditions, geometrical parameters, ⋯ can be considered as extra-coordinates. Then, by solving once the multidimensional resulting model, we have access to the space-time evolution for each value of the parameters that were introduced as extra-coordinates. The interested reader can refer to [11,14,29] and the references therein. This revisits this issue in the last section of the present paper.

Remark 7 Because we have decoupled in the solution algorithm the space and time problems, the meshes used for solving each one of the problems becomes uncorrelated. Thus, there are not stability constraints on the time step. Moreover, we could consider extremely small time steps without affecting the computation cost significantly, because that choice only affects the solution accuracy of the one-dimensional initial value problem serving to the calculation of functions $T_i(t)$.

Remark 8 Because the just argued decoupling, the problems that must be solved within the PGD framework at each step (the ones concerning the calculation of $R(x)$, $S(y)$ and $W(t)$) can be solved, if desired, by using different discretization methods for each one of them.

Remark 9 When the diffusivity becomes too small, the non-symmetry of the time differential operator requires a variant of the algorithm described above. In that case we should proceed to the residual minimization [10].

3. Numerical results

Before performing some numerical test in 2D and 3D, we are focusing in the thermal source that will be considered and the issues related to its space-time separated representation.

3.1. Thermal source

We consider a thermal source moving along the surface $y = H$ with a velocity v. Because in many industrial applications such thermal source consists of a laser beam, we assume that the thermal flux on the upper surface is modeled from a Gaussian distribution whose characteristic length will be denoted by l. In the 3D solutions addressed later, we will assume without loss of generality that this distribution is uniform in the z-direction. Thus, the thermal flux reads:

$$q(x,t) = A\cdot\frac{1}{l\sqrt{2\pi}}\cdot\exp\left(-\frac{(x-vt)^2}{2l^2}\right) \tag{34}$$

where A represents the thermal flux intensity.

In order to perform a separated representation description of $q(x,t)$ we compute the matrix $\mathbf{q}$ with components $\mathbf{q}_{j,r} = q(\tilde{x}_j, \tilde{t}_j)$, where $(\tilde{x}_j, \tilde{t}_j)$ are related to a corse mesh consisting of $\tilde{M}$ nodes on the upper boundary $y = H$ and $\tilde{P}$ time steps.

As soon as matrix $\mathbf{q}$ is defined, we can apply a singular value decomposition – SVD – that allows to define its separated form representation on the coarse mesh

$$q(\tilde{x},\tilde{t}) \approx \sum_{i=1}^{i=Q} \tilde{F}_i(\tilde{x})\cdot\tilde{G}_i(\tilde{t}) \tag{35}$$

By performing a projection of the functions involved in that representation on the fine calculation mesh, we obtain finally

$$q(x,t) \approx \sum_{i=1}^{i=Q} F_i(x)\cdot G_i(t) \tag{36}$$

When applying this procedure on the thermal flux (34) for $v = 0.1$, $A = 10^4$ and $l = 0.05$ (all units in the metric system) the separated representation consisting of the $Q = 15$ most significant functions $F_i(x)$ and $G_i(t)$ exhibits an approximation error of 0.03% when comparing the reconstructed solution (36) depicted in Fig. 1 with its exact expression (34). For the application of the SVD, a coarse mesh consisting of $\tilde{M} = 100$ nodes in the x-direction and $\tilde{P} = 100$ in the time axis was considered (even if it can be applied efficiently on the finer mesh). The functions that resulted from the SVD application were projected on the fine calculation mesh consisting of $M = 1000$ and $P = 1000$.

When considering sharper thermal sources the number of required terms for approximating it up to a certain accuracy increases in a significant manner. For example when considering a moving step, all the modes are relevant and no reduction can be made by applying the SVD. In those cases one could proceed without performing a space-time separated representation of the thermal source. If we observe the terms affected by such choice in the procedure described in the previous section we notice that in its first step (the one related to the calculation of function $R(x)$ (Eq. (26))) the boundary integral writes:

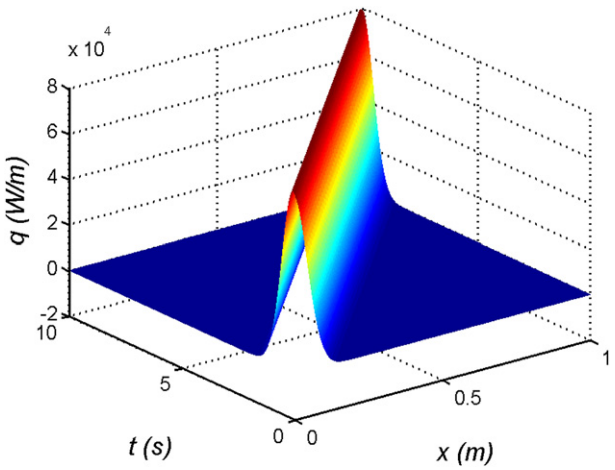

Fig. 1. Reconstructed heat flux consisting of a separated representation involving 15 terms.

$$\int_{\Omega_x\times\mathcal{I}} R^*\cdot S(y=H)\cdot W\cdot q(x,t)\mathrm{d}x\cdot\mathrm{d}t \tag{37}$$

that could be integrated numerically in the time interval $\mathcal{I}$.

In the second step, the one leading to the calculation of $S(y)$ it results:

$$\int_{\Omega_x\times\mathcal{I}} R\cdot S^*(y=H)\cdot W\cdot q(x,t)\mathrm{d}x\cdot\mathrm{d}t \tag{38}$$

to be integrated in $\Omega_x\times\mathcal{I}$, and finally in the third step, the one leading to the calculation of $W(t)$ it results

$$\int_{\Omega_x\times\mathcal{I}} R\cdot S(y=H)\cdot W^*\cdot q(x,t)\mathrm{d}x\cdot\mathrm{d}t \tag{39}$$

that must be integrated in Ω_x.

When the thermal sources are localized in space, $q(x,t)$ vanishes in the most part of the domain $\Omega_x\times\mathcal{I}$ and in that case previous integrals can be performed without major difficulties in a reasonable time. On the other hand it can be noticed that when the thermal source can be separated, integrations can be carried out very fast because multidimensional integrals can be computed from the product of one-dimensional integrals.

In the previous analysis the number of time steps $\tilde{P}$ of the coarse mesh used for separating the source term, $q(x,t)$, can be viewed as the number of snapshots in a POD construction of the space-time separated representation of the source term. It is important to check the convergence of the separated representation obtained with respect to the value of $\tilde{P}$ as well as the separability, ensured if the number of terms in the finite sum verifies $Q \ll \tilde{P}$.

3.2. 2D numerical test

In this section we are considering the geometry and the process conditions sketched in Fig. 2. The calculation mesh consists of 1000 nodes in the length, 100 nodes in the thickness and 1000 time steps. We consider the previous Gaussian flux with again $A = 10^4$, $v = 0.1$ and $l = 0.05$. The material thermal properties are $\rho = 1000\ \mathrm{kg\ m^{-3}}$, $C_p = 1000\ \mathrm{J\ kg^{-1}\ K^{-1}}$, $k_x = 5\ \mathrm{W\ K^{-1}\ m^{-1}}$ and $k_y = 0.5\ \mathrm{W\ K^{-1}\ m^{-1}}$. The initial temperature is set to $u^0 = 0\ ^\circ\mathrm{C}$ in the whole domain and the temperature on the boundary defined by $(y = 0)$ is constrained to

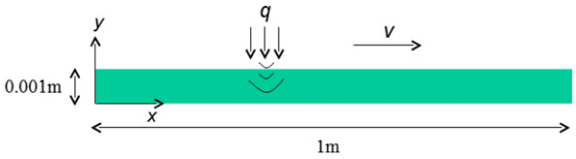

Fig. 2. Sketch of the geometry and process conditions.

$u(x,y=0,t)=0$ °C during the process. It is also considered in this test that there is no loss of heat on Γ_4 and then $\Gamma_4^s=\Gamma_4$ in Eq. (17).

Fig. 3 depicts the most relevant functions involved in the separated representation of the temperature field $X_i(x)$, $Y_i(y)$ and $T_i(t)$, $i=1,\cdots,4$. The reconstructed solution obtained from these functions is depicted in Fig. 4 at different times that correspond to different positions of the thermal source moving on the surface $y=H=0.001$.

When comparing the PGD solution with an equivalent finite element solution, considering the last one as the reference one, we noticed that the error (using a space-time L^2 norm) was lower than 1%. Thus, we can conclude on the efficiency of the proposed strategy that proceeds with impressive computing time savings without compromising the solution accuracy.

3.3. 3D numerical test

The procedure detailed above can be easily extended to 3D geometries. For that purpose it suffices to consider the space-time separated representation of the temperature field in a hexahedral tape.

$$u(x,y,z,t)\approx\sum_{i=1}^{N}X_i(x)\cdot Y_i(y)\cdot Z_i(z)\cdot T_i(t) \tag{40}$$

that is constructed by a simple extension of the iteration procedure described previously.

To prove the feasibility of such extension to higher dimensional models we consider the geometry addressed in the previous 2D example extruded in the z-direction with a depth of $0.2m$. Thus, $(x,y,z)\in\Omega$, $\Omega=\Omega_x\times\Omega_y\times\Omega_z$, with $\Omega_x=(0,L=1m)$, $\Omega_y=(0,H=0.001m)$ and $\Omega_z=(0,D=0.2m)$. We consider that the thermal flux does not vary in the z-direction such that the expression previously considered remains valid:

$$q(x,z,t)=A\cdot\frac{1}{l\sqrt{2\pi}}\cdot\exp\left(-\frac{(x-vt)^2}{2l^2}\right) \tag{41}$$

whose separated representation reads again:

$$q(x,z,t)\approx\sum_{i=1}^{i=Q}F_i(x)\cdot H_i(z)\cdot G_i(t) \tag{42}$$

with $H_i(z)=1,\ \forall i$.

We consider 1000, 100 and 1000 nodes for discretizing Ω_x, Ω_y and Ω_z respectively. Due to the uniformity of the solution in the z-direction, a very coarse discretization in that direction suffices, but we prefer to consider a mesh fine enough to highlight the capabilities of PGD and the interest of the coordinates separation. We consider as previously 1000 time steps. Of course, if one wants to solve the same problem using a standard mesh based discretization technique, the resulting model contains $1000\times100\times1000$, i.e. 10^8 nodes (degrees of freedom), and then, in the general case of non-linear material models one must solve 1000 times a system of size 10^8, that is practically intractable.

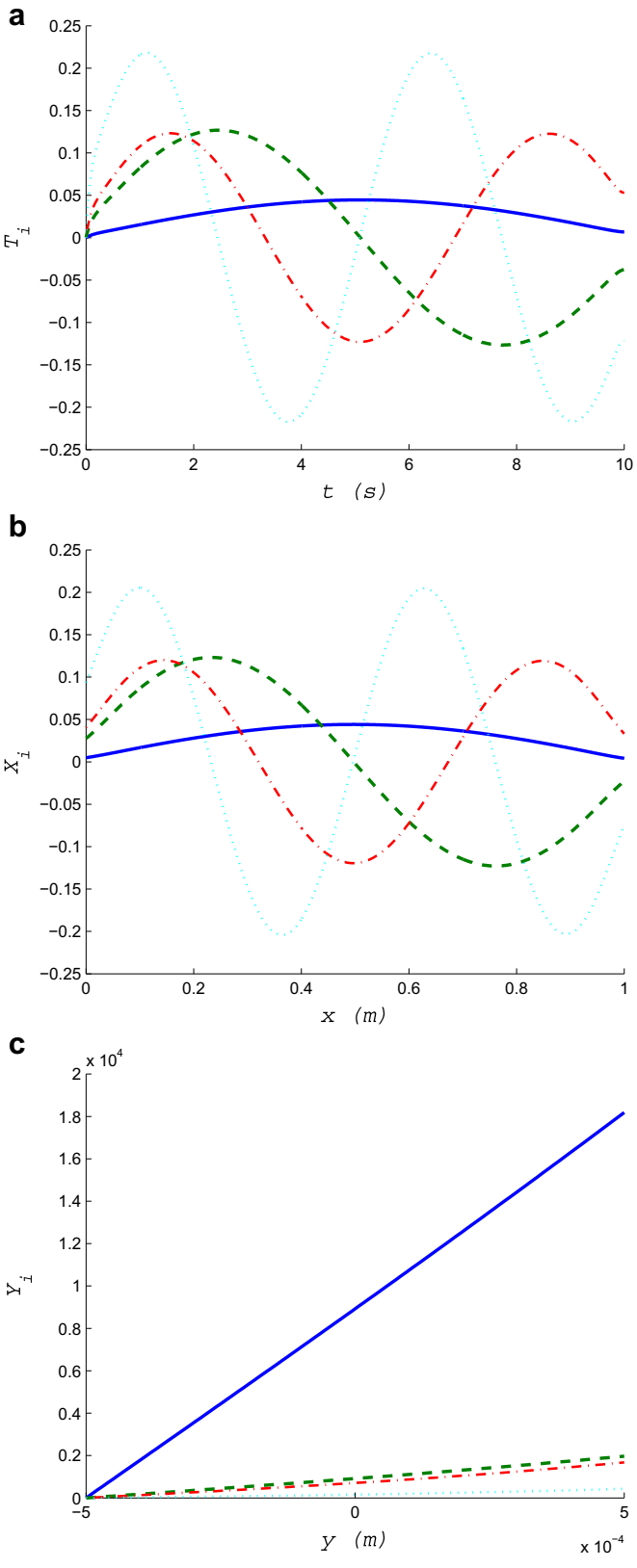

Fig. 3. Most significant functions involved in the separated representation of the temperature field.

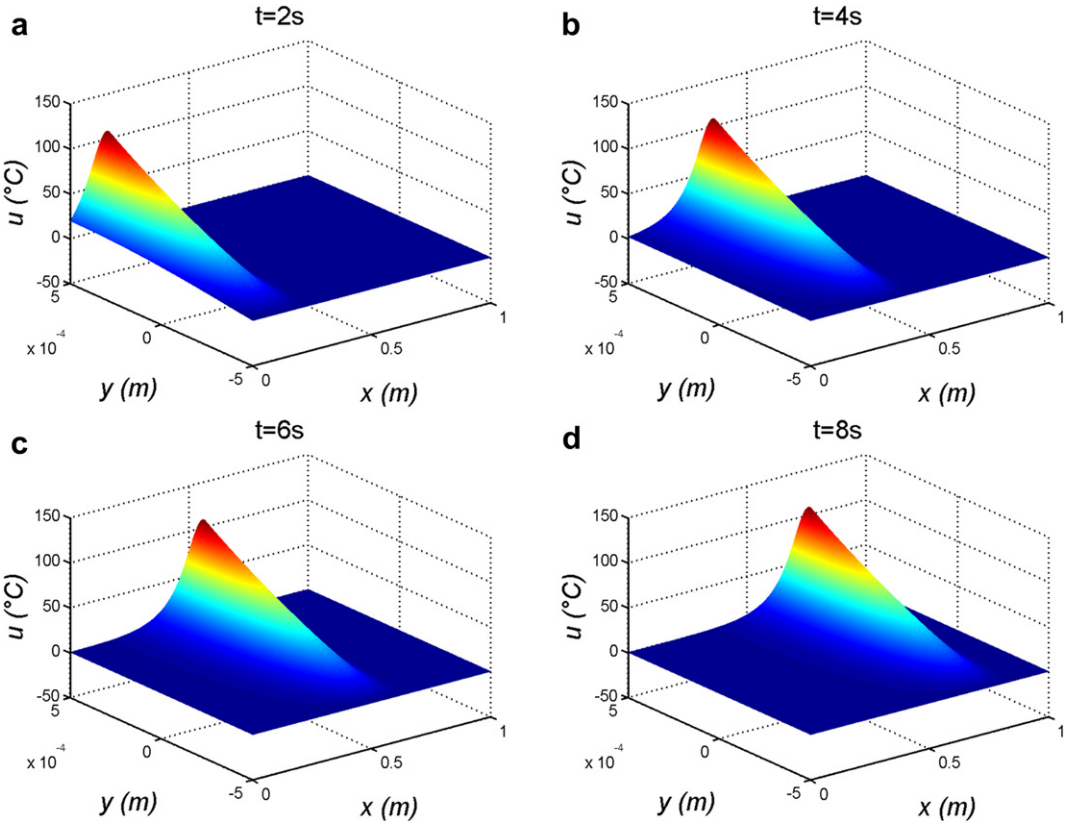

Fig. 4. Reconstructed thermal field at different times obtained from the separated representation whose functions are depicted in Fig. 3.

By using the PGD this solution is computed in around 1 min by using Matlab on a standard laptop. Instead of solving 1000 times, a systems of size 10^8 we must solve of the order of N 1D problems of size 1000 (leading to tridiagonal matrices) for computing functions $X_i(x)$ $(i = 1,\cdots,N)$, of the order of N 1D problems of size 100 for computing $Y_i(y)$, of the order of N 1D problems of size 1000 for calculating $Z_i(z)$ and of the order of N 1D initial value problems for computing $T_i(t)$. These calculation can be performed incredibly fast even in the non-linear case [4].

Fig. 5 depicts the reconstructed solution, where for visualization purposes we represented the temperature field (using a color map) on different sections along the tap thickness, without respecting the geometrical scale.

4. Towards an enhanced parametric modeling

Because the computational complexity when using separated representations increases linearly with the number of dimensions, instead the exponential increase characteristic of mesh-based discretization strategies, model parameters or the ones involved in the definition of the geometry or the initial and boundary conditions can be considered as extra-coordinates. In [21] a panoply of parameters where introduced as extra-coordinates in thermal homogenization. The interested reader can refer to [12] for a recent complete review. Then, by solving the resulting multi-dimensional model only once, we have access to the solution for any value of those parameters. In this circumstances optimization and inverse identification could be speeded-up many orders of magnitude as proved in our former works cited below, because for each tentative value of these parameters the solution is obtained by a simple particularization of the general parametric solution.

Optimization and inverse analysis were successfully accomplished in [14,15,17]. In the context of the applications considered in the present work we considered the source intensity A in Eq. (34) and its velocity v as extra-coordinates, because like this one could optimize the process trying to increase the velocity, implying an increase of the source intensity in order to ensure the thermal treatment, while avoiding a degradation of the material because an overexposure. In order to simplify the problem representation we

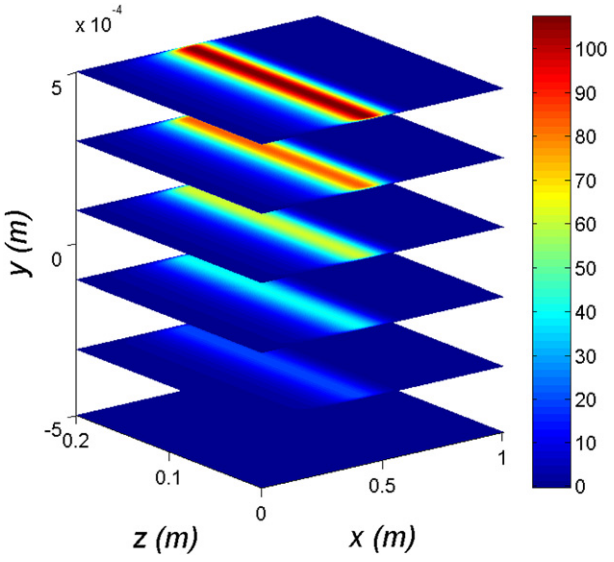

Fig. 5. Reconstructed temperature field at $t = 5s$.

decided in [27] to consider the reference system moving with the thermal source, and then an advective term appears in the heat equation, that now reads:

$$\rho \cdot C_p \cdot \left(\frac{\partial u}{\partial t} - v \cdot \nabla u \right) - \nabla \cdot (\mathbf{K} \cdot \nabla u) = 0 \tag{43}$$

When v is constant, thermal histories can be computed from the steady state solution of Eq. (43) with

$$\begin{cases} \left. \dfrac{\mathrm{d}u}{\mathrm{d}x} \right|_{\mathbf{x} \in \Gamma_1} = 0 \\ \left. \dfrac{\mathrm{d}u}{\mathrm{d}y} \right|_{\mathbf{x} \in \Gamma_2} = 0 \\ u(\mathbf{x} \in \Gamma_3) = u^0 \\ -k_y \cdot \left. \dfrac{\mathrm{d}u}{\mathrm{d}y} \right|_{\mathbf{x} \in \Gamma_4} = \begin{cases} -q(x,t); & x \in \Gamma_4^s \\ h \cdot (u(y = H, t) - u_{\mathrm{amb}}); & x \in \Gamma_4 - \Gamma_4^s \end{cases} \end{cases} \tag{44}$$

with now a fixed thermal source

$$q(x,t) = A \cdot \frac{1}{l\sqrt{2\pi}} \cdot \exp\left(-\frac{x^2}{2l^2} \right) \tag{45}$$

The solution of the previous problem was searched in the parametric form $u(x,y,A,v)$ by using the separated representation

$$u(x,y,A,v) \approx \sum_{i=1}^{i=N} X_i(x) \cdot Y_i(y) \cdot v_i(A) \cdot V_i(v) \tag{46}$$

Now, if we focus in a particular location, the solution can be computed as a function of the process parameters A and v:$u(A,v;x,y)$. The interested reader can refer to [15] and [27] for some interesting applications in control of processes, the last reference concerning the parametric modeling just described.

5. Conclusion

In this paper we addressed the issue of fully space-time separated representations of thermal models defined in tape-type domains. These degenerate geometries are more and more considered in composite forming processes justifying the interest for fast and accurate simulations of processes, especially in the case of tricky process conditions involving moving thermal sources applied on the domain boundary.

We proposed a fully separated representation that transforms a three dimensional transient problem into a sequence of 4 one dimensional ones. The computing time savings can be simply spectacular, allowing the solution of models never solved until now due to the extremely large number of degrees of freedom.

The use of the PGD opens a number of unimaginable possibilities, some of them are being explored, others are waiting for deeper analysis.

References

[1] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids, Journal of Non-Newtonian Fluid Mechanics 139 (2006) 153–176.

[2] A. Ammar, D. Ryckelynck, F. Chinesta, R. Keunings, On the reduction of kinetic theory models related to finitely extensible dumbbells, Journal of Non-Newtonian Fluid Mechanics 134 (2006) 136–147.

[3] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. Part II: transient simulation using space-time separated representation, Journal of Non-Newtonian Fluid Mechanics 144 (2007) 98–121.

[4] A. Ammar, M. Normandin, F. Daim, D. Gonzalez, E. Cueto, F. Chinesta, Non-incremental strategies based on separated representations: applications in computational rheology, Communications in Mathematical Sciences 8/3 (2010) 671–695.

[5] A. Ammar, F. Chinesta, P. Diez, A. Huerta, An error estimator for separated representations of highly multidimensional models, Computer Methods in Applied Mechanics and Engineering 199 (2010) 1872–1880.

[6] R.A. Bialecki, A.J. Kassab, A. Fic, Proper orthogonal decomposition and modal analysis for acceleration of transient FEM thermal analysis, International Journal for Numerical Methods in Engineering 62 (2005) 774–797.

[7] T. Bui-Thanh, K. Willcox, O. Ghattas, B. van Bloemen Waanders, Goal-oriented, model-constrained optimization for reduction of large-scale systems, Journal of Computational Physics 224/2 (2007) 880–896.

[8] J. Burkardt, M. Gunzburger, H-Ch. Lee, POD and CVT-based reduced-order modeling of Navier–Stokes flows, Computer Methods in Applied Mechanics and Engineering 196 (2006) 337–355.

[9] F. Chinesta, A. Ammar, F. Lemarchand, P. Beauchene, F. Boust, Alleviating mesh constraints: model reduction, parallel time integration and high resolution homogenization, Computer Methods in Applied Mechanics and Engineering 197/5 (2008) 400–413.

[10] F. Chinesta, A. Ammar, E. Cueto, Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models, Archives of Computational Methods in Engineering 17/4 (2010) 327–350.

[11] F. Chinesta, A. Ammar, A. Leygue, R. Keunings, An overview of the proper generalized decomposition with applications in computational rheology, Journal of Non-Newtonian Fluid Mechanics 166 (2011) 578–592.

[12] F. Chinesta, P. Ladeveze, E. Cueto, A short review in model order reduction based on proper generalized decomposition, Archives of Computational Methods in Engineering 18 (2011) 395–404.

[13] M. Girault, E. Videcoq, D. Petit, Estimation of time-varying heat sources through inversion of a low order model built with the Modal Identification Method from in-situ temperature measurements, International Journal of Heat and Mass Transfer 53 (2010) 206–219.

[14] Ch. Ghnatios, F. Chinesta, E. Cueto, A. Leygue, P. Breitkopf, P. Villon, Methodological approach to efficient modeling and optimization of thermal processes taking place in a die: application to pultrusion, Composites Part A 42 (2011) 1169–1178.

[15] Ch. Ghnatios, F. Masson, A. Huerta, E. Cueto, A. Leygue, F. Chinesta, Proper generalized decomposition based dynamic data-driven control of thermal processes, Computer Methods in Applied Mechanics and Engineering, in press.

[16] D. Gonzalez, A. Ammar, F. Chinesta, E. Cueto, Advances in the use of separated representations, International Journal for Numerical Methods in Engineering 81/5 (2010) 637–659.

[17] D. Gonzalez, F. Masson, F. Poulhaon, A. Leygue, E. Cueto, F. Chinesta, Proper generalized decomposition based dynamic data-driven inverse identification, Mathematics and Computers in Simulation, in press.

[18] M.D. Gunzburger, J.S. Peterson, J.N. Shadid, Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data, Computer Methods in Applied Mechanics and Engineering 196 (2007) 1030–1047.

[19] P. Ladeveze, Nonlinear Computational Structural Mechanics, Springer, NY, 1999.

[20] P. Ladeveze, J.-C. Passieux, D. Néron, The LATIN multiscale computational method and the proper generalized decomposition, Computer Methods In Applied Mechanics and Engineering 199/21–22 (2010) 1287–1296.

[21] H. Lamari, A. Ammar, P. Cartraud, G. Legrain, F. Jacquemin, F. Chinesta, Routes for efficient computational homogenization of non-linear materials using the proper generalized decomposition, Archives of Computational Methods in Engineering 17/4 (2010) 373–391.

[22] Y. Maday, E.M. Ronquist, The reduced basis element method: application to a thermal fin problem, SIAM Journal on Scientific Computing 26/1 (2004) 240–258.

[23] D. Néron, P. Ladevèze, Proper generalized decomposition for multiscale and multiphysics problems, Archives of Computational Methods in Engineering 17/4 (2010) 351–372.

[24] S. Niroomandi, I. Alfaro, E. Cueto, F. Chinesta, Real-time deformable models of non-linear tissues by model reduction techniques, Computer Methods and Programs in Biomedicine 91 (2008) 223–231.

[25] S. Niroomandi, I. Alfaro, E. Cueto, F. Chinesta, Order Reduction for hyperelastic materials, International Journal for Numerical Methods in Engineering 81/9 (2010) 1180–1206.

[26] H.M. Park, D.H. Cho, The use of the Karhunen-Loève decomposition for the modelling of distributed parameter systems, Chemical Engineering Science 51 (1996) 81–98.

[27] F. Poulhaon, Ch. Ghnatios, B. Bogner, A. Barasinski, F. Chinesta, A. Leygue, A numerical approach for the evaluation of the residual stresses in the automated tape placement process. In the proceedings of the ASME 2012 11th Biennial Conference on Engineering Systems Design and Analysis, 2012.

[28] C. Prud'homme, D.V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. Patera, G. Turinici, Reliable real-time solution of parametrized partial differential equations: reduced-basis output bound methods, Journal of Fluids Engineering 124 (2002) 70–80.

[29] E. Pruliere, F. Chinesta, A. Ammar, On the deterministic solution of parametric models by using the proper generalized decomposition, Mathematics and Computers in Simulation 81 (2010) 791–810.

[30] D. Ryckelynck, A priori hyper-reduction method: an adaptive approach, Journal of Computational Physics 202 (2005) 346–366.

[31] D. Ryckelynck, L. Hermanns, F. Chinesta, E. Alarcon, An efficient “a priori” model reduction for boundary element models, Engineering Analysis with Boundary Elements 29 (2005) 796–801.
[32] D. Ryckelynck, F. Chinesta, E. Cueto, A. Ammar, On the a priori model reduction: overview and recent developments, Archives of Computational Methods in Engineering 13/1 (2006) 91–128.
[33] K. Veroy, A. Patera, Certified real-time solution of the parametrized steady incompressible Navier–Stokes equations: rigorous reduced-basis a posteriori error bounds, International Journal Numerical Methods in Fluids 47 (2005) 773–788.
[34] E. Videcoq, O. Quemener, M. Lazard, A. Neveu, Heat source identification and on-line temperature control by a branch eigenmodes reduced model, International Journal of Heat and Mass Transfer 51 (2008) 4743–4752.